Econometriscs assignment

Heidi Marie Rolfsnes and Ann Elisabeth Jacobsen

# MSB104 Econometrics

This is a collection of assignments in the subject econometrics MSB104.

#Assignemnt 1 In this assignment we are looking at five European countries. Denmark, France, Hungary, Portugal and Slovakia. In the first assignment we have collected data from Eurostats, the data contains GDP and populations for the countries but also at NUTS3 level for the last 20 years.

We collected 2 datasets, and the first thing we did was to clean the list and remove data that didn’t have any valuables. We changed the names on some of the variables. After that we merged the two datasets into one.

Then we calculated the GDP pr capita. And this we did by dividing the GDP on the population.

We have divided the countries into NUTS3 levels. To get the best possible results, the countries are in their own chunk.

##Denmark

Denmark are missing observations fore some of the years. so there are less data to be collected about then, Denmark is also a small country so there are also not that many NUTS2 regions.

## # A tibble: 11 × 15  
## # Groups: Regio\_id [11]  
## Regio…¹ `2007` `2008` `2009` `2010` `2011` `2012` `2013` `2014` `2015` `2016`  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 DK011 59142. 59920. 58406. 64695. 62798. 64038. 64435. 67811. 68399. 69961.  
## 2 DK012 57784. 62389. 59913. 63842. 64276. 66999. 69980. 71440. 77034. 77484.  
## 3 DK013 34652. 36008. 34540. 34706. 34121. 35599. 36513. 37285. 39472. 40137.  
## 4 DK014 29627. 29458. 28832. 29695. 30582. 31785. 32120. 34177. 34883. 35091.  
## 5 DK021 30940. 29326. 30884. 31511. 31778. 31943. 32148. 32683. 34003. 35349.  
## 6 DK022 29576. 29712. 27606. 29276. 29543. 30815. 30781. 31454. 33473. 33193.  
## 7 DK031 33851. 35385. 34109. 34074. 34894. 35113. 35521. 36360. 37082. 38406.  
## 8 DK032 41451. 42289. 40229. 42700. 43808. 44689. 45970. 47187. 48071. 48989.  
## 9 DK041 41802. 42297. 40105. 40309. 42430. 42255. 43465. 44913. 43977. 48361.  
## 10 DK042 38086. 39318. 38505. 38602. 39430. 39881. 39921. 40566. 41339. 44136.  
## 11 DK050 37145. 37815. 36400. 36862. 37301. 38522. 38953. 39377. 40529. 41654.  
## # … with 4 more variables: `2017` <dbl>, `2018` <dbl>, `2019` <dbl>,  
## # `2020` <dbl>, and abbreviated variable name ¹​Regio\_id

## `summarise()` has grouped output by 'Regio\_id'. You can override using the  
## `.groups` argument.

## # A tibble: 154 × 3  
## # Groups: Regio\_id [11]  
## Regio\_id Year GDP\_per\_Capita  
## <chr> <dbl> <dbl>  
## 1 DK011 2007 59142.  
## 2 DK011 2008 59920.  
## 3 DK011 2009 58406.  
## 4 DK011 2010 64695.  
## 5 DK011 2011 62798.  
## 6 DK011 2012 64038.  
## 7 DK011 2013 64435.  
## 8 DK011 2014 67811.  
## 9 DK011 2015 68399.  
## 10 DK011 2016 69961.  
## # … with 144 more rows

summary(DK)

## Year Regio\_id Per\_capita   
## Min. :2007 Length:154 Min. :27606   
## 1st Qu.:2010 Class :character 1st Qu.:35172   
## Median :2014 Mode :character Median :40004   
## Mean :2014 Mean :44209   
## 3rd Qu.:2017 3rd Qu.:48033   
## Max. :2020 Max. :86083

##France

France is a large country, it has over a hundred NUTS3 regions.

## # A tibble: 87 × 20  
## # Groups: Regio\_id [87]  
## Regio…¹ `2002` `2003` `2004` `2005` `2006` `2007` `2008` `2009` `2010` `2011`  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 FR101 75369. 75357. 76586. 79258. 80305. 85155. 85583. 81539. 87627. 88674.  
## 2 FR102 21927. 22509. 23447. 23712. 24979. 26127. 27454. 26935. 26395. 28112.  
## 3 FR103 30762. 31098. 31614. 32814. 33740. 35640. 37549. 35666. 36894. 37946.  
## 4 FR104 28957. 30139. 30786. 30331. 31953. 33756. 36314. 34861. 35827. 34777.  
## 5 FR105 71406. 71787. 74158. 78255. 80136. 85353. 89494. 86217. 92362. 93365.  
## 6 FR106 26270. 26760. 27908. 29004. 30428. 31919. 32814. 31705. 33961. 33656.  
## 7 FR107 26120. 27437. 28327. 28770. 30842. 32699. 33018. 33822. 35112. 35247.  
## 8 FR108 24146. 24706. 26156. 27109. 27905. 29539. 30822. 29457. 29668. 30419.  
## 9 FRB01 21684. 21533. 22569. 22827. 23856. 25061. 24372. 23789. 23793. 24115.  
## 10 FRB02 21592. 21417. 21881. 22283. 22848. 23844. 24232. 23257. 23521. 24460.  
## # … with 77 more rows, 9 more variables: `2012` <dbl>, `2013` <dbl>,  
## # `2014` <dbl>, `2015` <dbl>, `2016` <dbl>, `2017` <dbl>, `2018` <dbl>,  
## # `2019` <dbl>, `2020` <dbl>, and abbreviated variable name ¹​Regio\_id

## `summarise()` has grouped output by 'Regio\_id'. You can override using the  
## `.groups` argument.

## # A tibble: 1,630 × 3  
## # Groups: Regio\_id [87]  
## Regio\_id Year GDP\_per\_Capita  
## <chr> <dbl> <dbl>  
## 1 FR101 2002 75369.  
## 2 FR101 2003 75357.  
## 3 FR101 2004 76586.  
## 4 FR101 2005 79258.  
## 5 FR101 2006 80305.  
## 6 FR101 2007 85155.  
## 7 FR101 2008 85583.  
## 8 FR101 2009 81539.  
## 9 FR101 2010 87627.  
## 10 FR101 2011 88674.  
## # … with 1,620 more rows

summary(FR)

## Year Regio\_id Per\_capita   
## Min. :2002 Length:1630 Min. : 8292   
## 1st Qu.:2006 Class :character 1st Qu.: 22050   
## Median :2011 Mode :character Median : 24476   
## Mean :2011 Mean : 26869   
## 3rd Qu.:2016 3rd Qu.: 28270   
## Max. :2020 Max. :116235

##Hungary

## # A tibble: 20 × 20  
## # Groups: Regio\_id [20]  
## Regio…¹ `2002` `2003` `2004` `2005` `2006` `2007` `2008` `2009` `2010` `2011`  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 HU110 14449. 15013. 17088. 19097. 19739. 21939. 23699. 21452. 21946. 22268.  
## 2 HU120 6297. 6693. 7429. 8192. 8082. 9649. 9563. 8185. 8777. 8696.  
## 3 HU211 6604. 6951. 8047. 8510. 8818. 9644. 10225. 8029. 8727. 9572.  
## 4 HU212 6495. 7816. 9159. 10347. 9515. 11038. 11365. 9286. 10048. 10309.  
## 5 HU213 5630. 5924. 6465. 6780. 6742. 7644. 7986. 6597. 7261. 7263.  
## 6 HU221 8492. 9089. 9621. 10217. 10929. 12129. 12815. 10581. 12125. 12639.  
## 7 HU222 6863. 7887. 8322. 8551. 9144. 9320. 9368. 7921. 8606. 9163.  
## 8 HU223 6140. 6989. 7593. 7743. 7416. 8209. 8903. 7662. 8584. 8223.  
## 9 HU231 5225. 5505. 6023. 6325. 6375. 7089. 7431. 6419. 6630. 6561.  
## 10 HU232 4790. 5154. 5572. 5867. 5774. 6283. 6697. 5957. 6323. 6529.  
## 11 HU233 5593. 5296. 5788. 6099. 6060. 7065. 7825. 7084. 7338. 7649.  
## 12 HU311 4387. 4649. 5361. 6097. 5984. 6535. 6649. 5582. 5942. 6169.  
## 13 HU312 5163. 5377. 5790. 6170. 6241. 7160. 7483. 6354. 6809. 6878.  
## 14 HU313 3773. 3979. 4193. 4417. 4458. 4478. 4810. 4114. 4395. 4541.  
## 15 HU321 5509. 5970. 6563. 6936. 6816. 7287. 7753. 6906. 7452. 7692.  
## 16 HU322 4646. 4877. 5307. 5615. 5913. 6331. 6859. 6150. 6071. 6646.  
## 17 HU323 4145. 4342. 4693. 5030. 4936. 5329. 5599. 4949. 5376. 5651.  
## 18 HU331 5062. 5203. 5838. 6288. 6303. 6864. 7467. 6389. 6626. 6994.  
## 19 HU332 4575. 4698. 5231. 5413. 5378. 6015. 6324. 5307. 5719. 6034.  
## 20 HU333 5665. 5850. 6474. 6934. 6952. 7426. 8036. 7078. 7350. 7470.  
## # … with 9 more variables: `2012` <dbl>, `2013` <dbl>, `2014` <dbl>,  
## # `2015` <dbl>, `2016` <dbl>, `2017` <dbl>, `2018` <dbl>, `2019` <dbl>,  
## # `2020` <dbl>, and abbreviated variable name ¹​Regio\_id

## `summarise()` has grouped output by 'Regio\_id'. You can override using the  
## `.groups` argument.

## # A tibble: 380 × 3  
## # Groups: Regio\_id [20]  
## Regio\_id Year GDP\_per\_Capita  
## <chr> <dbl> <dbl>  
## 1 HU110 2002 14449.  
## 2 HU110 2003 15013.  
## 3 HU110 2004 17088.  
## 4 HU110 2005 19097.  
## 5 HU110 2006 19739.  
## 6 HU110 2007 21939.  
## 7 HU110 2008 23699.  
## 8 HU110 2009 21452.  
## 9 HU110 2010 21946.  
## 10 HU110 2011 22268.  
## # … with 370 more rows

summary(HU)

## Year Regio\_id Per\_capita   
## Min. :2002 Length:380 Min. : 3773   
## 1st Qu.:2006 Class :character 1st Qu.: 6330   
## Median :2011 Mode :character Median : 7754   
## Mean :2011 Mean : 8782   
## 3rd Qu.:2016 3rd Qu.: 9907   
## Max. :2020 Max. :31013

##Slovenia

## # A tibble: 8 × 20  
## # Groups: Regio\_id [8]  
## Regio\_id `2002` `2003` `2004` `2005` `2006` `2007` `2008` `2009` `2010` `2011`  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 SK010 11478. 12994. 15023. 18295. 20452. 25868. 29882. 30925. 32670. 33733.  
## 2 SK021 4850. 5817. 6821. 7876. 10433. 12693. 14131. 13031. 14154. 14704.  
## 3 SK022 4500. 5191. 6038. 6453. 7938. 9654. 11064. 10431. 11050. 11520.  
## 4 SK023 4045. 4773. 5648. 6420. 7218. 8665. 10329. 9896. 10282. 11587.  
## 5 SK031 3970. 4469. 5233. 6059. 6881. 8766. 10667. 10315. 11328. 11462.  
## 6 SK032 4122. 4706. 5204. 5205. 6094. 7534. 8966. 8452. 9114. 9119.  
## 7 SK041 2977. 3343. 3818. 4281. 4549. 5637. 7090. 6804. 7125. 7690.  
## 8 SK042 4382. 4932. 5653. 6176. 7079. 8531. 9967. 9167. 9888. 10209.  
## # … with 9 more variables: `2012` <dbl>, `2013` <dbl>, `2014` <dbl>,  
## # `2015` <dbl>, `2016` <dbl>, `2017` <dbl>, `2018` <dbl>, `2019` <dbl>,  
## # `2020` <dbl>

## `summarise()` has grouped output by 'Regio\_id'. You can override using the  
## `.groups` argument.

## # A tibble: 152 × 3  
## # Groups: Regio\_id [8]  
## Regio\_id Year GDP\_per\_Capita  
## <chr> <dbl> <dbl>  
## 1 SK010 2002 11478.  
## 2 SK010 2003 12994.  
## 3 SK010 2004 15023.  
## 4 SK010 2005 18295.  
## 5 SK010 2006 20452.  
## 6 SK010 2007 25868.  
## 7 SK010 2008 29882.  
## 8 SK010 2009 30925.  
## 9 SK010 2010 32670.  
## 10 SK010 2011 33733.  
## # … with 142 more rows

summary(SK)

## Year Regio\_id Per\_capita   
## Min. :2002 Length:152 Min. : 2976   
## 1st Qu.:2006 Class :character 1st Qu.: 7829   
## Median :2011 Mode :character Median :11057   
## Mean :2011 Mean :12493   
## 3rd Qu.:2016 3rd Qu.:13655   
## Max. :2020 Max. :40095

## Portugal

Finely we have a look at Portugal.

i### Descriptive Statistics GDP per capita & Population

**Country level**

## GDP\_per\_Capita  
## 1 14415.77

**Regional level (NUTS3)**

## # A tibble: 456 × 21  
## Regio\_id Population `2002` `2003` `2004` `2005` `2006` `2007` `2008` `2009`  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 PT111 250491 8954. NA NA NA NA NA NA NA   
## 2 PT111 250823 NA 9058. NA NA NA NA NA NA   
## 3 PT111 250476 NA NA 9345. NA NA NA NA NA   
## 4 PT111 250134 NA NA NA 9775. NA NA NA NA   
## 5 PT111 249616 NA NA NA NA 10447. NA NA NA   
## 6 PT111 248983 NA NA NA NA NA 10865. NA NA   
## 7 PT111 248515 NA NA NA NA NA NA 11333. NA   
## 8 PT111 247492 NA NA NA NA NA NA NA 11629.  
## 9 PT111 246690 NA NA NA NA NA NA NA NA   
## 10 PT111 245306 NA NA NA NA NA NA NA NA   
## # … with 446 more rows, and 11 more variables: `2010` <dbl>, `2011` <dbl>,  
## # `2012` <dbl>, `2013` <dbl>, `2014` <dbl>, `2015` <dbl>, `2016` <dbl>,  
## # `2017` <dbl>, `2018` <dbl>, `2019` <dbl>, `2020` <dbl>

### Descriptive Statistics regional inequity (Gini Nuts2)

```{r Portugal gini #| echo = FALSE #Nuts 2 id PT %>%  
mutate(id\_nuts2 = substr(Regio\_id,1,4)) #Nuts 1 id PT %>%  
mutate(id\_nuts0 = substr(Regio\_id,1,2))

#gen weighted gini #for all gini.wtd(PTPopulation)

PT %>% group\_by(id\_nuts2) %>% mutate(gini\_n2 = gini.wtd(Per\_capita, w = Population)) %>% ungroup() # to remove grouping

summary(PT$gini\_n2 )

```